

Realidad Aumentada en Secuencias de Vídeo

Nil Domene Esteban

Resumen— Los coches autónomos son un tema de actualidad ya que se espera que en los próximos años revolucionen la forma de transporte. Para que este proceso tenga éxito, es necesario que se dispongan de todas las herramientas que permitan alcanzar la capacidad técnica necesaria y que además garanticen la máxima seguridad. El Deep Learning ha permitido el uso de nuevas técnicas que han aumentado el ritmo de los progresos en este sector. Esta tecnología requiere una gran variedad de datos para entrenamiento y prueba que deben de abarcar todo tipo de situaciones de conducción. En este trabajo se propone una nueva aproximación a la hora de abarcar la mayor diversidad de datos posible mediante la utilización de realidad aumentada en secuencias de vídeo. Permitiendo así que una única secuencia pueda dar lugar a una gran variedad de situaciones que quizá hubieran sido imposible de obtener mediante la toma de datos real. Esta nueva aproximación rompe con lo visto hasta ahora ya que proporciona una gran flexibilidad a la forma de obtener datos que no se ha explorado hasta estos últimos años. Esta forma de generar datos se explora a continuación mediante el uso de datos obtenidos a través de LIDAR y a técnicas de posproducción.

Palabras clave— Visión Por Computador, LIDAR, Reconstrucción 3D, Nube de Puntos, Conducción Autónoma, Realidad Aumentada.

Abstract— Autonomous cars are a trending topic since it is expected they completely change the way we move. For this project to be successful, it is essential to have the means to achieve the technical capacity and guarantee the maximum level of safety. Deep Learning has allowed the use of new technologies which have accelerated the pace of the sector progress. This technology needs a broad spectrum of data to train and test, encompassing the wide number of situations which are found in the real environment. This paper proposes a new approach to improve the data collection making use of augmented reality in video sequences, allowing one single sequence to provide a wide range of situations which would perhaps have been impossible to collect from the real environment. This new approach, unexplored until recent years, breaks with what has been seen before, as it provides an enormous flexibility to data collection. This new approach is explored below by means of data collected from LIDAR and post-production techniques.

Keywords— Computer Vision, LIDAR, 3D Reconstruction, Point Cloud, Autonomous Driving, Augmented Reality.



1 INTRODUCCIÓN

Hoy en día se están produciendo grandes cambios en la generación de algoritmos que permiten realizar tareas que hasta ahora habían sido consideradas como imposibles para ser abarcadas mediante la inteligen-

cia artificial. Uno de estos problemas había sido siempre la conducción autónoma. La forma de adquirir datos en este campo hasta hace poco se basaba en conducir un vehículo en un entorno real e ir almacenando todos los datos que se van obteniendo a través de los distintos sensores instalados en el vehículo. Esta estrategia que se ha convertido en la referencia, pero lleva asociados una serie de problemáticas como la poca flexibilidad para poder conseguir someter a un vehículo a un conjunto exhaustivo de pruebas en las que se encuentren tanto escenas cotidianas de conducción como otras que contengan escenarios con alguna particularidad. Este hecho pone de manifiesto que se deben buscar nuevas maneras de adquirir los datos y si nos basamos únicamente

- E-mail de contacto: Nil.Domene@e-campus.uab.cat
- Mención realizada: Ingeniería de Computación
- Trabajo tutorizado por: Antonio Manuel López Peña (Departamento de Ciencias de la Computación)
- Curso 2018/19

en los métodos tradicionales de adquisición de datos es muy probable que no lleguemos a saber nunca cual es el comportamiento del sistema, en situaciones que no se hayan podido captar en la creación del dataset.

Un problema adicional que plantea esta forma de adquisición de los datos es la monotonía de las secuencias que se capturan. Ya que los sucesos excepcionales suceden muy pocas veces y por tanto resulta muy complicado que se acabe obteniendo una gran cantidad de datos muy parecidos provocando que la diversidad de los datos sea muy limitada.

Otro problema importante que presentan estas estrategias basadas en la adquisición de datos mediante las estrategias clásicas son los altos costes asociados. Se necesita una gran cantidad de horas de vídeo para disponer de una cierta diversidad en los datos de entrada. Además deben de considerarse los costes indirectos que provienen de la conducción como el consumo de gasolina, el coste del conductor, etc. Otro restricción asociada a este enfoque es la cantidad de tiempo requerida para la creación de un nuevo conjunto de datos sobre el que trabajar. Además del tiempo requerido para la toma de datos se le debe sumar la preparación de las secuencias mediante un proceso de selección y clasificación.

Aunque puede parecer que la única forma de generar nuevos datos sería la acumulación de más secuencias de vídeo mediante el recorrido de kilómetros adicionales, pueden existir alternativas a este enfoque que permitan la generación de largos datasets a partir de la generación de nuevos contenidos a partir de un conjunto de datos reducido.

El punto que se explora aquí, es el uso de estrategias de generación de datos distintas a las usadas hasta ahora. Realizando una valoración sobre si estas nuevas aproximaciones nos van a permitir mejorar la capacidad de creación de nuevos datasets.

Debido a las limitaciones que presentan las formas actuales de abordar al problema se han buscado nuevas formas de trabajar. Dentro de estas nuevas estrategias, destacan aquellas basadas en métodos híbridos que usan las secuencias originales junto con nuevos elementos insertados de forma artificial. Esta nueva estrategia permite que aunque dispongamos de relativamente pocos datos obtenidos mediante el uso de las técnicas actuales, aumentemos el tamaño de los datasets y la diversidad de datos que contienen. En concreto en este trabajo se estudia la generación de nuevas secuencias de vídeo que integran los contenidos obtenidos a través de las cámaras insertadas en los vehículos y técnicas de realidad aumentada que permiten insertar nuevos objetos dentro de la escena.

Además en este proyecto se explora la forma de automatizar todo el proceso de generación de estas nuevas escenas, de forma que se necesite de la mínima interacción humana para la generación de estos nuevos datos.

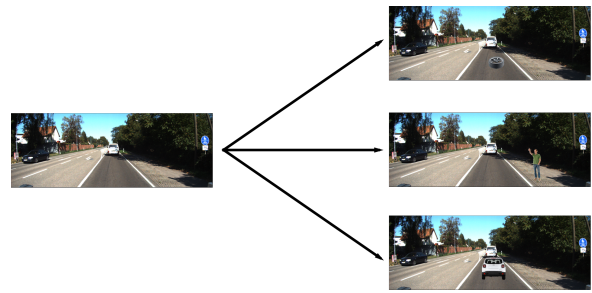


Fig. 1: Ejemplo de las posibilidades que ofrece el uso de realidad aumentada para generar nuevas secuencias a partir de una secuencia original.

Aunque esta nueva estrategia pueda resultar muy atractiva también presenta una serie de problemas. El más significativo de ellos es intentar realizar una reconstrucción coherente de la escena original que nos permita insertar elementos en una localización que se corresponda con su análoga en el vídeo original. Este problema se puede abarcar mediante el uso de estrategias SLAM y datos LIDAR para realizar una representación del aspecto de la escena y el recorrido dentro de ella. Aunque esta estrategia haga imprescindible el uso de la tecnología LIDAR, las posibilidades que ofrece esta tecnología superan a las alternativas que se podrían plantear mediante el uso únicamente de la cámara.

Uno de los puntos más destacados de esta nueva forma en la creación de datos es la facilidad con la que se pueden crear nuevos datasets que contengan nuevas realidades de las que puede ser que no existan datos en ese momento como los patinetes eléctricos ya que su irrupción en los últimos años ha sido muy notoria y desconocemos como va a comportarse el vehículo autónomo en presencia de esta clase de vehículos. Por tanto puede resultar ser una muy buena forma de generar nuevos datasets de forma ágil que permitan afrontar con rapidez los cambios que se van produciendo y valorar como se comportan los vehículos en presencia de ellos.

Con este proyecto se busca explorar las posibilidades ofrecidas por esta nueva estrategia basada en la reconstrucción de una única escena original mediante los datos obtenidos a través de los sensores y las posibles creaciones que se pueden generar de forma automática a partir de ellos. Una de las problemáticas asociadas a la generación de estos datos era el tiempo que requerían. Ya que se intenta solventar el problema del tiempo requerido, resulta fundamental automatizar el proceso de creación de estos nuevos datasets en el mayor grado posible.

Los nuevos datos generados en este proyecto se basaran en la inserción de distintos tipos de objetos en las escenas generadas que se puedan encontrar en una conducción real como son nuevos coches que antes no aparecían, nuevos peatones en las zonas circundantes de la vía o nuevas señales de tráfico.

2 OBJETIVOS

Este proyecto busca relacionar conceptos de Visión por Computación junto con otros a nivel artísticos para dar lugar a un vídeo con realidad aumentada. Se busca además

valorar distintas alternativas sobre la reconstrucción de la escena, y realizar un análisis de las herramientas que van a permitir llevar a cabo este proyecto. Es también imprescindible hacer una valoración objetiva sobre el rendimiento final del proyecto y los factores que limitan su aplicación directa a otras escenas. Uno de los puntos importantes de este proyecto es realizar la mayor automatización posible para que sea fácil empezar a trabajar en la versión con realidad aumentada de cualquier escena de la que se dispongan de los datos necesarios para realizar su reconstrucción. De forma resumida en este proyecto se busca abarcar los siguientes conceptos para crear el resultado final:

1. Reconstruir la escena captada mediante los datos captados por el LIDAR y el uso de técnicas SLAM.
2. Aplicar una iluminación precisa en los nuevos objetos insertados.
3. Automatizar todo el proceso en el mayor grado posible.

Para realizar todos los puntos comentados ha sido necesario realizar un proceso de selección de las herramientas que se van a usar en el desarrollo. El trabajo aquí presentado no está pensado para trabajar con una única secuencia, en cambio busca la máxima flexibilidad posible y así poder permitir trabajar con cualquier secuencia que el usuario desee. Los puntos mencionados son los pilares fundamentales sobre los que se produce el desarrollo de este software. La escena reconstruida debe de mantener el mismo aspecto que la original, haciendo posible distinguir todos los elementos de la secuencia original para que las inserciones de los nuevos objetos se produzcan en el lugar correcto.

También se busca usar una técnica de SLAM que contenga un nivel de error relativamente bajo para que la trayectoria quede perfectamente insertada dentro del 3D reconstruido y no varíe mucho con respecto a la trazada en la secuencia original. Si el nivel de error del método de localización es elevado repercute directamente en la calidad final de la nueva secuencia ya que la proyección de todos los objetos será incorrecta.

Otro punto crucial es conseguir una iluminación correcta en toda la escena. Si no se consigue emular correctamente la luz que se encuentra en el vídeo los elementos que se inserten presentarán un aspecto antinatural y destacarán más de lo que deberían. Gracias a la iluminación correcta los nuevos elementos van a presentar un aspecto similar a los contenidos originalmente y pueden llegar a pasar desapercibidos.

El último punto es desarrollar un software que sea sencillo de utilizar y que requiera de la mínima interacción humana para poder generar un resultado. Además el proyecto busca presentarse como una alternativa basada en el código libre proveyendo de todo el código fuente generado y además usando herramientas gratuitas para todos y cada uno de los procesos.

3 MÉTODOS

Para poder realizar la reconstrucción son imprescindibles una serie de datos que permitan crear el equivalente tridimensional de la escena y su posterior versión con realidad aumentada. Los datos requeridos se pueden observar en la figura 2 que transmite a nivel global aquello que es imprescindible para que el software funcione:

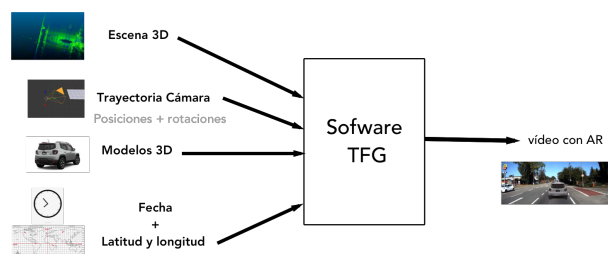


Fig. 2: Esquema general acerca del funcionamiento a nivel global del proyecto. Se muestran los inputs que necesita para poder reconstruir la escena y el output que es un vídeo que contiene AR.

Escena 3D Es necesario disponer de una reconstrucción 3D de la escena para la generación de estos nuevos vídeos. Los datos necesarios para este propósito se obtienen a partir del sensor LIDAR que llevan incorporado muchos de los vehículos autónomos. Estos van a servir para reconstruir el aspecto tridimensional original que presenta la escena mediante la captura del aspecto geométrico de entorno a lo largo de todo el tiempo que dura la secuencia. Mediante este tipo de estrategias se puede situar correctamente un objeto en un lugar determinado y que mantenga la coherencia con el vídeo original que se ha grabado. Por ejemplo se puede colocar un objeto en un andén y que luego al reconstruir la escena se mantenga en el sitio.

Trayectoria cámara Es necesaria una reconstrucción de la trayectoria de la cámara dentro de un espacio tridimensional, ya que esta equivale a la trayectoria trazada por el vehículo durante la grabación de la escena. Un punto importante aquí es comprender que la extracción de estas posiciones no es trivial y se necesitan de métodos complejos para poder extraer la trayectoria realizada. Aquí se hace uso de estrategias conocidas como SLAM que se encargan de extraer las posiciones en las que se encuentra situada la cámara mediante la extracción de características en el vídeo. Algunos datasets además disponen del Ground Truth de la trayectoria de la cámara obtenida a través de los sensores inerciales que además sirven de referencia para comparar con los resultados obtenidos por las técnicas de localización. Esto es lo que se conoce como odometría y es un problema recurrente en la conducción autónoma.

Modelos 3D Para poder crear nuevas escenas a partir del vídeo original es imprescindible disponer de una serie de modelos tridimensionales que se van a disponer en un lugar concreto dentro de la escena reconstruida y así poder generar la apariencia de que se encontraba en la original. La inserción de cada uno de estos modelos cambia comple-

tamente el aspecto de la escena, generando una secuencia distinta para cada una de las inserciones.

Fecha junto con latitud y longitud Para poder generar una escena realista es imprescindible que cuente con una iluminación coherente con el contenido original. Es necesario por tanto situar los elementos de iluminación en una localización y dirección de proyección que se ajusten al vídeo original. Para conseguirlo es necesario el uso de la latitud y longitud además de la fecha y la hora para situar una iluminación global que coincida con la posición en la que se encuentra localizado el Sol.

Uno de los resultados que se obtiene en este proyecto es un archivo Python que al ser importado en Blender va a permitir trabajar de manera directa en la escena reconstruida para la inserción de los modelos que se deseen por parte del usuario. Una vez se haya realizado este paso se procede a la obtención del resultado final que es un vídeo en realidad aumentada que es la unión entre la escena original que llega al sistema y el vídeo generado en Blender.

La elaboración del resultado final implica el uso de una serie de herramientas que pueden verse resumidas en la figura 3:



Fig. 3: Software empleado durante el desarrollo del proyecto.

En la figura 4 se puede observar un esquema general sobre como esta construido todo el proyecto. El elemento principal que necesitamos es un dataset que esté diseñado para el ámbito de la conducción autónoma. Aparte de la selección del conjunto de datos con el que vamos a trabajar debemos escoger una secuencia en particular que va a ser usada para generar N variantes. Después se debe completar el fichero de configuración donde especificaremos una serie de parámetros acerca de la escena sobre la que se va a trabajar. Una vez se han especificado estas configuraciones se produce la reconstrucción de los distintos elementos que se encuentran dentro de la escena. Estos son el entorno en el que ha sido captada la secuencia y la reconstrucción de la trayectoria que ha recorrido el vehículo dentro de ella. Una vez se han realizado estos dos puntos, se procede a la creación de la nueva escena mediante la integración de todos estos elementos más la inserción del elemento adicional que queramos incluir dentro de la secuencia. En este momento lo único que nos falta para dar por concluido todo el proceso es interpolar la nueva secuencia generada más la secuencia original de la que partíamos para dar lugar a la nueva secuencia que va a contener la realidad aumentada.

3.1. Datasets

Un punto muy important a la hora de realizar este trabajo es la selección de un dataset que permita desarrollar todo

los objetivos que se han establecido. Para que una escena cualquiera pueda ser empleada por este software debería contar con la secuencia de vídeo que se ha obtiene en la conducción, más los datos LIDAR asociados y los datos sobre la localización y la fecha en la que se grabó la escena. Como el proyecto se centra en la generación de nuevos datasets para la conducción autónoma, es imprescindible que las secuencias que se usen para generar nuevas que incluyan la realidad aumentada pertenezcan también a este ámbito.

Este hecho descarta algunos datasets que contengan escenas en movimiento pero que no estén especialmente enfocados a la conducción autónoma, también se descartan así las grabaciones caseras basadas en el uso de la cámara Kinect. Aunque algunos de estos datasets cumplan con la información necesaria para realizar la reconstrucción no se ajustaban al marco de este trabajo.

También se ha evitado usar datasets de conducción autónoma que no incluyeran datos LIDAR, ya que sin estos la estrategia de reconstrucción posterior no tiene sentido como podría haber sucedido en caso de utilizar el dataset de Berkeley[1] o Baidu Apollo Scape [2]. Estos únicamente estaban formados para secuencias de vídeo y están pensados para trabajar más en problemas de segmentación o similares más que en la extracción de la odometría de la escena.

Otro punto a destacar es que aunque no todas las secuencias que se encuentran dentro de un mismo datasets son igual de válidas para la reconstrucción ya que aquellas que implican objetos en movimiento dentro de la escena dificultan la reconstrucción 3D aparte de que es mucho más difícil realizar una inserción coherente dentro de una escena que contenga muchos elementos móviles.

Esto se debe al hecho de que las zonas que queden ocultas momentáneamente por un objeto en movimiento dentro de la escena, en su análogo tridimensional van a ser una nube de puntos dispersa que representan todo el rastro que va dejando el vehículo mientras ha sido escaneado por el LIDAR. Por tanto en la reconstrucción es complicado saber en que punto exactamente del desplazamiento se encuentra el objeto para aplicar la opacidad de la forma correcta ya que se muestra como un rastro que abarca todo el movimiento que ha realizado.

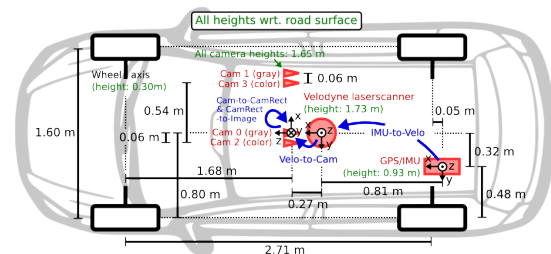


Fig. 5: Configuración del sistema empleado en KITTI para la captura de datos. Se pueden observar la disposición de los distintos sensores colocados en el vehículo.

El dataset KITTI está orientado al análisis de muchos problemas distintos todos ellos relacionados con la conducción autónoma. Este hecho favorece a que esté formado por gran variedad de secuencias en las que se abarcan muchas situaciones distintas. Estas secuencias están sepa-

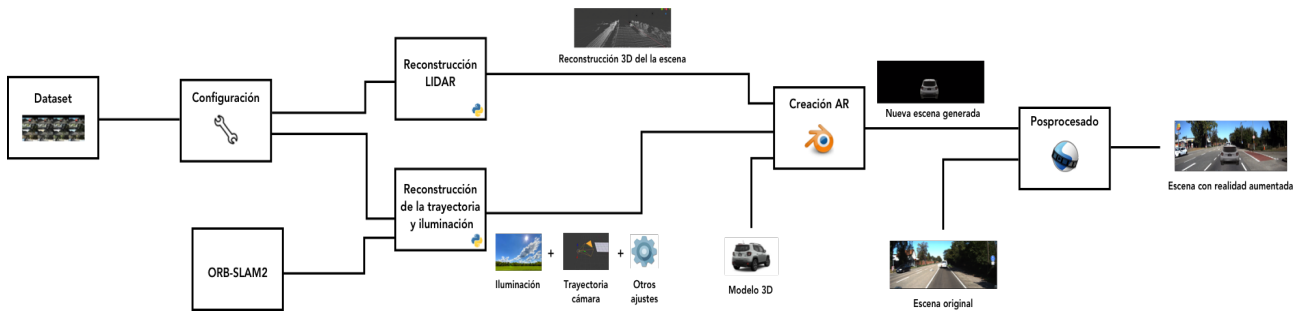


Fig. 4: Esquema general acerca del funcionamiento a nivel global del proyecto. Se muestran los inputs que necesita para poder reconstruir la escena y el output que es un vídeo que contiene AR.

radar en cuatro conjuntos (además de uno pensado para la calibración y otro centrado en el movimiento de una única persona), donde intentan abordar distintos tipos de entornos habituales en la conducción. Estas agrupaciones de secuencias son: **City** que contiene la mayor variedad de secuencias donde estas van desde calles estrechas a carreteras interurbanas, **Residential** que está centrada en calles estrechas como las que se pueden encontrar en urbanizaciones, **Road** formada únicamente por secuencias en carretera largas donde además que contienen tráfico, y por último también tenemos **Campus** que está centrada en zonas donde hay gran afluencia de peatones. Las secuencias que se van a usar pertenecen al conjunto **Residential** ya que incluyen secuencias donde hay poco movimiento en las zonas adyacentes a la carretera, haciéndolas perfectas para situar objetos en los andenes de las secuencias. Aunque se ha seleccionado KITTI debido a que es uno de los datasets más importantes que hay en el ámbito de la conducción autónoma, también es importante destacar que podrían haberse usado otros como el dataset de Málaga [3] o el Ford Campus [4] donde ambos cumplen con todos los requisitos necesarios para poder realizar este proyecto.

4 RECONSTRUCCIÓN DE LA ESCENA MEDIANTE LIDAR

La reconstrucción de la escena debe ser lo más correcta posible para poder situar en ella todos los elementos adicionales deseados y que presenten a su vez una coherencia. En este módulo escrito en Python se usan todas las nubes de puntos de las que dispone la secuencia para formar otras nubes de puntos construidas como una agregación de estas. Dicho de otra forma, se unen una serie de N nubes distintas (el usuario define esta cantidad) que dan lugar a una nube de puntos que las contiene a todas ellas. En vez de usar una única nube de puntos global se ha decidido usar M nubes de puntos donde cada grupo equivale a un fragmento de la escena. Esta estrategia mejora notablemente el aspecto de la reconstrucción si se compara con el resultado aportado por un único frame. Para realizar este proceso se hace uso de un algoritmo conocido ICP [5] que es el encargado de alinear pares de nubes de puntos. La estrategia que se sigue aquí es coger las dos primeras nubes de puntos de cada grupo y unir las, después se cogen las otras nubes de puntos de forma secuencial y se van uniendo con la nube de puntos generada por las dos primeras. El proceso puede verse resumido en la siguiente figura:

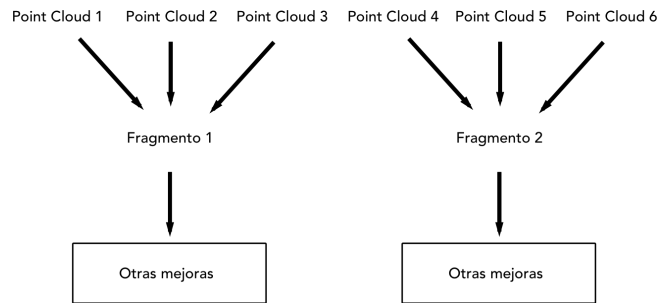


Fig. 6: Esquema general del módulo donde N nubes de puntos obtenidas directamente a través del LIDAR se entrelazan para formar una única. Se pueden realizar mejoras posteriores como subsampling o el cálculo de las normales.

Esta estrategia no es la que originalmente se planteó, ya que en un principio lo que se buscaba es realizar la reconstrucción total de la escena en una única nube de puntos que las contuviera a todas, este proceso se descartó debido a dos motivos. El primero es la alta sensibilidad que tiene ICP para organizar las nubes de puntos ya que resulta que si se usan nubes de puntos que contengan elementos móviles perjudican enormemente el resultado final convirtiendo la nube generada en un conjunto de nubes superpuestas sin ningún sentido. El segundo motivo se debe a los límites técnicos asociados a estas nubes de puntos ya que rápidamente pasaban a contar con decenas de millones de puntos haciéndolas inmanejables a la hora de trabajar con otro software.

Otra línea que fue investigada al principio fue la creación de modelos tridimensionales que se correspondieran a las nubes de puntos para poder trabajar con mayor facilidad los aspectos relacionados con la iluminación y las oclusiones dentro de la escena. Para poder conseguir esto se procedió a realizar un primer intento al uso de una única nube de puntos junto con una técnica conocida como la reconstrucción de Poisson [6] para comprobar si el aspecto que presentaba reproducía con fidelidad la escena.

El resultado obtenido en este caso reportaba una calidad pésima aun habiendo probado decenas de configuraciones distintas para ver el impacto que tenían, también aquí se realizaron modificaciones sobre las nubes para ver si se podía mejorar de alguna forma el aspecto que presentaba. Después de afrontar esta problemática se decidió probar otra estrategia siguiendo la misma línea pero en vez de aplicar la reconstrucción a una única nube de puntos se basaba en la idea de conseguir juntar la mayor cantidad de nubes posibles usando la misma estrategia del ICP.

Esta estrategia tampoco resultó ya que la geometría que re-

portaba no era el resultado esperado ya que al incluir una mayor cantidad de puntos totales que en un principio deberían incrementar muy positivamente la estética a nivel global, incluía mucho más ruido en toda la escena retornando unos resultados al mismo nivel que la estrategia basada en una única nube. En este punto, se probó a incrementar el nivel de suavizado de la reconstrucción de Poisson mediante el ajuste de los distintos parámetros que se tienen que introducir de manera arbitraria.

Este procedimiento retorna unos resultados aún peores ya que las estructuras geométricas más pronunciadas como farolas o coches presentaba una geometría muy distinta con el resto de su entorno se veían altamente perjudicados. A continuación se pueden observar algunos de los resultados reportados por estas estrategias:

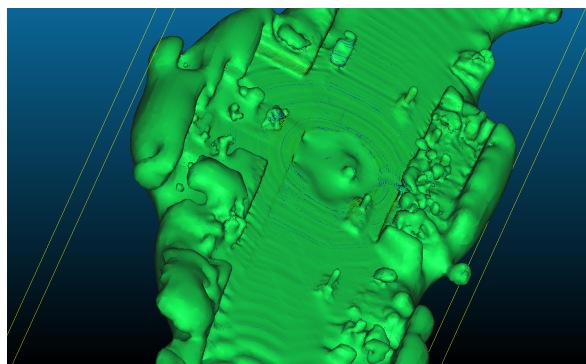


Fig. 7: Reconstrucción de la escena usando un único frame junto con técnicas de suavizado para mejorar el aspecto.

En este punto se decidió cambiar de rumbo y abandonar esta estrategia en pro de usar directamente estas nubes de puntos dentro de Blender y ver las posibilidades que se ofrecían. Un punto esencial que permite el correcto funcionamiento de esta herramienta es el uso de CloudCompare que a través de una herramienta de línea de comandos permite realizar muchas funcionalidades. El módulo usa la línea de comandos y realiza llamadas indicando una serie de parámetros. Primero se especifican los dos ficheros con los que se va a proceder a trabajar a continuación, después se indican como se va a trabajar. En el caso particular de este proyecto indicamos que queremos usar el algoritmo ICP y indicamos con un parámetro opcional el número de puntos que van a ser usados para la evaluación del error total que se produce en la alineación.

Este cálculo únicamente se basa en una ponderación entre todas las distancias entre pares de puntos de ambas nubes y la configuración resulta esencial en la obtención de las nubes de puntos con las alineaciones correctas. Si no se usa un número de puntos relativamente elevado es posible que el resultado no sea el deseado y las nubes queden mal alineadas. Una vez terminado el proceso de alineación, se procede a unir las nubes resultantes con el objetivo de tener una única nube que va a ser la que se va a exportar posteriormente. Además se procede a definir el formato de exportación que se va a usar para el resultado que en el caso de este proyecto es directamente formato binario. Otro punto a favor del uso de formato binario es el hecho de que soportado Blender y permite trabajar con el con mucha más facilidad.

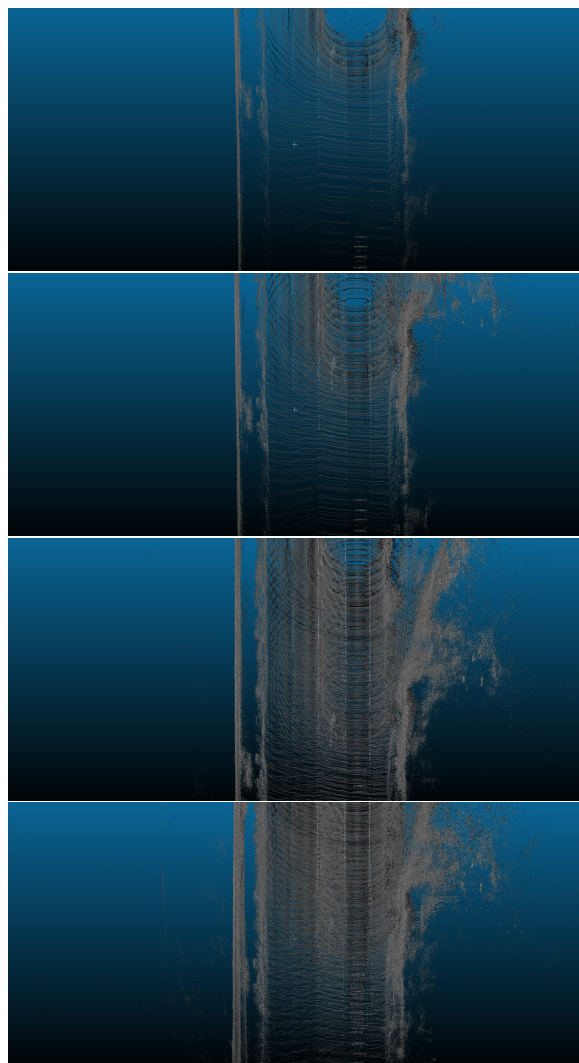


Fig. 8: Cambios de densidad de las nubes generadas para un único frame en la primera imagen y las mejoras que se aportan mediante el uso de ICP al unir 2, 5 y 10 frames respectivamente.

5 CREACIÓN DE LOS ELEMENTOS DE LA ESCENA

En este segundo módulo se procede a la generación del script que va a contener la trayectoria de la cámara, ajustes varios del sistema de renderizado y aspectos de la iluminación que posteriormente va a ser importado en Blender. Para la trayectoria de la cámara se procede al uso de ficheros que contienen las posiciones y rotaciones para cada uno de los frames y se procede a leerlos secuencialmente. Estos datos de la cámara se pueden obtener de dos maneras distintas, en el caso de que la secuencia posea además un Ground Truth de la odometría se puede usar directamente los valores que se proporcionan. En caso de no disponer de estos datos es necesario realizar una llamada al software ORB_SLAM2 para que nos genere este nuevo fichero con el mismo formato que el que tenemos en el groundtruth.

El directorio en el que se producen todas las lecturas debe ser especificado en el fichero de configuración de la misma manera que si deseamos o no usar el software de localización o si por el contrario vamos a usar la trayectoria real seguida por el vehículo. Una vez se han realizado las lecturas

de todos los ficheros y se han extraído todas las posiciones y rotaciones de la cámara en un espacio tridimensional, se procede a calcular cual es la posición real en la que se encuentra el sol en la escena original. Para poder resolver este problema se hace uso tanto de la fecha en que fue grabado como la longitud y latitud del primer fotograma de vídeo. Una vez realizado esto mediante el uso de una librería de Python externa [7] que está implementada directamente en base a los trabajos de Ibrahim Reda y Afshin Andreas [8] y nos va a facilitar el cálculo en coordenadas esféricas del Sol en base al radio de la tierra. La librería anterior nos reporta la altura y el ángulo azimut en el que se encuentra el Sol. Estos datos necesitan ser traducidos a su posición análoga en x, y, z para su inserción dentro de la escena reconstruida.

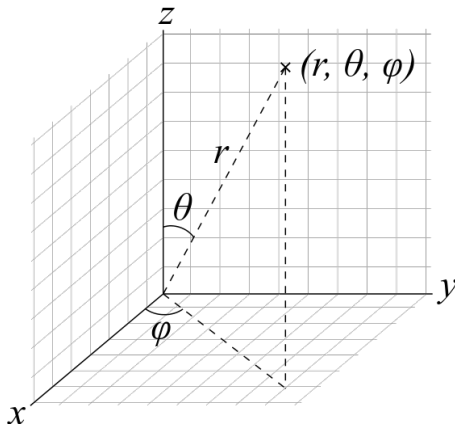


Fig. 9: Relación entre un espacio de coordenadas esféricas y cartesianas

Las fórmulas usadas para obtener las nuevas posiciones mediante la transformación de coordenadas esféricas a cartesianas:

$$\theta = \arccos\left(\frac{h}{r}\right)$$

$$x = r \sin \theta \cos \phi$$

$$y = r \sin \theta \sin \phi$$

$$z = r \cos \theta$$

Es imprescindible volver a calcular la altura de este en base a un radio máximo más reducido con el objetivo de situar la iluminación mucho más cerca de los nuevos elementos insertados y se aprecien mejor los detalles.

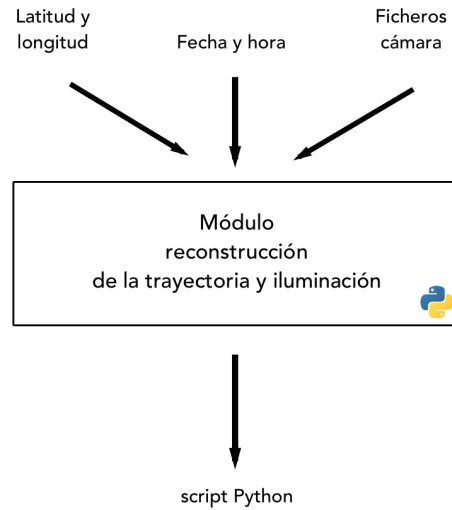


Fig. 10: Representación general del funcionamiento del módulo que genera el script que debe ser importado luego en Blender.

6 DETALLES ADICIONALES

Aunque sería posible llegar a generar las secuencias de forma completamente automática, en este proyecto se decide delegar al usuario la selección del lugar y el movimiento que va a realizar el nuevo objeto insertado. Mediante técnicas de segmentación podría llegar a posicionarse correctamente cualquier objeto en una posición determinada, pero debido a la escasez de tiempo para desarrollar todo el proyecto no ha sido posible explorar esta alternativa. A pesar de esto en este proyecto se ha realizado la inserción de un objeto de manera automática para que el usuario ya tenga un objeto con el que poder trabajar.

Para empezar a crear la secuencia de realidad aumentada es necesario únicamente abrir el software Blender mediante la línea de comandos y especificar mediante un parámetro adicional que deseamos importar el script Python que se ha generado previamente. Si hemos realizado esto correctamente nos encontraremos con que Blender se abre que consta de la trayectoria de la cámara, una iluminación global en el sitio correcto, un objeto dentro de la escena y por último unos ejes de referencia vacíos. Estos ejes son los que se van a usar para poder trabajar con la nube de puntos, esto implica que se debe tener instalado el plugin adicional Point Cloud Visualizer [9], que permite importar las nubes de puntos de manera muy sencilla y que además permite realizar algunas operaciones sobre ellas dentro de Blender.

Una vez hayamos importado correctamente la nube de puntos, si todo se ha realizado correctamente deberíamos disponer una escena correctamente configurada sobre la que se puede empezar a trabajar. Cuando se haya configurado la escena, se procede a renderizar la animación únicamente con los detalles que deseamos añadir al vídeo original. Si se ha trabajado con elementos que necesiten de oclusiones y se hayan generado de manera artificial se debe proceder a asignarles una textura completamente negra y que no emita ningún tipo de reflejo. Si esto no se realiza correctamente es muy probable que al intentar aplicar el efecto croma surjan problemas. En la figura número 11 se puede observar los elementos que intervienen en este proceso:

Lo que se debe realizar a continuación es el último paso,

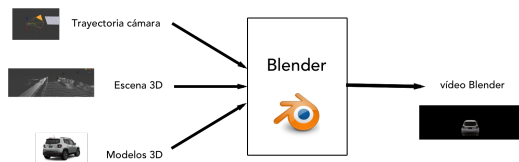


Fig. 11: Figura que resume el proceso realizado en Blender que mediante el uso del script va a ser capaz de generar una nueva secuencia con fondo transparente

debemos importar la escena original y el resultado obtenido en Blender con el objetivo de unir ambas en una secuencia que incorpore ambos contenidos. Lo único que se debe tener en cuenta en este punto es situar la nueva escena generada con el efecto croma para eliminar todo el fondo generado en la exportación de este y que esté colocada por encima en la jerarquía de secuencias que hay dentro del software OpenShot.

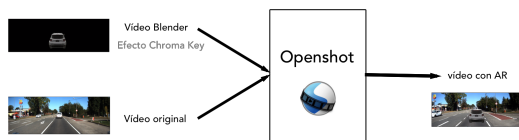


Fig. 12: Resumen del proceso realizado en OpenShot, donde a través del uso del efecto chroma key se pueden interpolar ambos vídeos.

7 RESULTADOS

Se han explorado muchas estrategias para intentar abordar el problema de reconstruir la escena con la mayor calidad gráfica posibles. La representación de los elementos adicionales dentro de la escena original presentan una coherencia con la iluminación real que incluía la escena y además se ha conseguido simplificar el proceso en el mayor grado posible con la intención de que su utilización sea lo más flexible posible.

El uso de un método de renderizado de vídeo distinto del empleado en la versión previa mejoró en gran medida la calidad final. Esta nueva versión incluía mejoras en las texturas de los nuevos objetos y empezaba a presentar un aspecto mucho más realista. En este punto se quiso exponer la flexibilidad que proporciona esta técnica ya que permite que con una misma escena modificando únicamente las texturas de alguno de los objetos que se encuentran dentro de ella, cambia completamente el aspecto que presenta el vídeo.

Como punto final en el proyecto se procedió a cambiar los objetos por otros que tuvieran texturas de alta definición, y además se incorporó el sistema de iluminación realista basada en la localización real del Sol. Esta última estrategia es la que ha presentado el mejor aspecto global y se ha dejado cómo muestra final del aspecto del proyecto.



Fig. 13: Variantes de un mismo vehículo insertado en la escena



Fig. 14: Escena con realidad aumentada con texturas de alta definición y un sistema de iluminación global mejorado

Aquí además se pueden observar algunas otras imágenes que muestran algunas de las posibilidades que se pueden alcanzar mediante este proyecto:

En la generación de resultados se han explorado distintos tipos de objetos para ver el aspecto final que presentaban una vez insertados en la escena. Estos nuevos objetos insertados eran tanto estáticos como en movimiento. Los que están insertados de forma fija en la escena son los que son más sensibles a los pequeños errores que se hayan producido en la reconstrucción. Si no se tiene la posición exacta de la cámara los objetos pueden llegar a desplazarse incorrectamente una distancia entre unos centímetros y unos metros en el peor de los casos. Aun teniendo las posiciones de la cámara se presentan pequeños errores debidos a las irregularidades de la carretera.

Por otro lado los objetos que contienen movimiento suavizan los errores que se puedan producir en la posición de la cámara. Este hecho se puede observar tanto en el uso de los Ground Truths pero es mucho más acentuado en los casos



Fig. 15: La primera imagen muestra el aspecto original que presentaba el fotograma. En la siguiente imagen se puede observar un nuevo cartel que aparece en la escena, y en la última imagen se puede observar la inserción de un peatón.

en los que se hayan usado métodos de reconstrucción de la trayectoria como puede ser los resultados obtenidos a través de ORB_SLAM2. Otro punto importante a comentar acerca del error es pequeño en las rectas, pero en las partes de la secuencia donde se tracen curvas el nivel de error crece. Este error también se ve afectado por el tiempo y las zonas de la secuencia que aparecen más tarde van a presentar unos niveles de error más elevados a pesar de las optimizaciones que se producen mediante el uso de loop closing y las técnicas de relocalización.

El siguiente punto importante a tratar sería el uso de una herramienta como 3DTK que está más preparada para trabajar con largas de secuencias de datos LIDAR y generar unas escenas 3D globales más precisas que la alternativa aquí presentada basada en el uso de fragmentos. Otro punto a mejorar en líneas futuras es la manera en la que se realiza la superposición del vídeo con AR a la escena original ya que el software empleado (OpenShot) presenta algunas dificultades a la hora de crear este efecto. En algunos de los objetos que contengan una parte sin iluminación o con una textura demasiado oscura sea detectado como fondo y sea eliminado de forma involuntaria.

Otro de los problemas que presenta Openshot es la forma en la que trata los bordes dentro del croma generando muchas veces un borde negro añadido al objeto que seguramente sea posible eliminar mediante el uso de otras herramientas. La forma que usa para realizar la extracción del fondo además evita la proyección correcta de las sombras proyectadas por los nuevos objetos insertados. Además del uso de esta herramienta se podría además automatizar de la misma manera que se ha realizado con CloudCompare, generando así la malla completa de la escena que posteriormente sería importada en Blender para trabajar con ella.

El proyecto aquí realizado se puede consultar en el siguiente

enlace: https://github.com/nilde/TFG_FINAL

8 CONCLUSIONES Y TRABAJO FUTURO

En el trabajo aquí presentado se muestra el resultado que se ha obtenido en la creación de escenas con realidad aumentada en secuencias de vídeo incluidas dentro del ámbito de la conducción autónoma. Se ha conseguido unir la reconstrucción 3D de la escena con un ligero post procesado para poder dar los resultados que se han podido observar en el apartado anterior. Los puntos más significativos a destacar una vez se ha terminado el proyecto es observar que aunque los resultados obtenidos sean satisfactorios no es posible realizar una herramienta de reconstrucción genérica que funcione a la perfección para todas las escenas.

Siempre se va a estar limitado por el nivel de movimiento de los objetos internos dentro de la escena y por la precisión de los métodos SLAM a través de los que se lleva a cabo la estimación de las posiciones en las que se va encontrando la cámara. Además cuando se ha finalizado el desarrollo se ha podido comprobar que disponemos de maneras muy creativas de trabajar con los datos de los que se disponen y que este nuevo tipo de técnicas basadas en la generación de escenas híbridas son una muy buena herramienta de cara a la investigación.

En futuras líneas de investigación de este proyecto se debería realizar una comparación exhaustiva entre distintos métodos de SLAM para poder comprobar si existe alguna mejor alternativa al que se acabó optar en este proyecto (ORB_SLAM2), y si las mejoras al usar otro realmente se aprecian en el aspecto final del vídeo.

También sería interesante usar un método de SLAM basado en el uso de datos LIDAR para la reconstrucción de la trayectoria, ya que este tipo de estrategias son por el momento el estado del arte a la hora de abordar este problema reportando las mejores puntuaciones para numerosos datasets incluido el escogido para realizar este proyecto.

Otro punto a realizar al finalizar este proyecto sería el insertar objetos animados dentro de la escena. Aunque ya se han realizado inserciones de objetos móviles dentro de la escena, estos únicamente son una traslación de un objeto estático dentro de la reconstrucción. Si se llegara a realizar la inserción de objetos que realmente se movieran dentro de la escena la calidad resultante del producto se elevaría. El último punto importante sobre el que se debería trabajar es la elaboración de una estrategia que elimine las partes de la nube de puntos que pertenezcan a objetos móviles en las secuencias originales.

Esto se podría llevar a cabo mediante el uso de estrategias de segmentación aplicadas a las nubes de puntos y a la posterior extracción de todos aquellos elementos que no formen parte de la carretera.

9 AGRADECIMIENTOS

Quiero dar las gracias a mi tutor Antonio M. López Peña que gracias a sus constantes consejos y sugerencias han hecho posible completar este proyecto.

REFERENCIAS

- [1] Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., and Darrell, T., “BDD100K: A diverse driving video database with scalable annotation tooling,” *CoRR*, vol. abs/1805.04687, 2018.
- [2] Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y., and Yang, R., “The ApolloScape Dataset for Autonomous Driving,” *CoRR*, vol. abs/1803.0, 2018.
- [3] Gonz, J., “The Malaga Urban Dataset : High-rate Stereo and Lidars in a realistic urban scenario,” vol. 2013, no. C, pp. 1–11, 2018.
- [4] Pandey, G., Eustice, R., and McBride, J., “Ford Campus Vision and Laser Data Set,” *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.
- [5] Bennewitz, M., Arras, K., and Robotics Book, P., “Iterative Closest Point Algorithm Introduction to Mobile Robotics,”
- [6] Kazhdan, M., Bolitho, M., and Hoppe, H., “Poisson Surface Reconstruction,” in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, (Aire-la-Ville, Switzerland, Switzerland), pp. 61–70, Eurographics Association, 2006.
- [7] S-bear, “Python code for computing sun position.”
- [8] Reda, I. and Andreas, A., *Solar position algorithm for solar radiation application*, vol. 76. may 2004.
- [9] SBVC, “Addon to import PLY files as point clouds into Blender.”
- [10] Community, B. O., “Blender – a 3D modelling and rendering package,” in <http://www.blender.org/>, (Stichting Blender Foundation, Amsterdam), 2018.
- [11] EDF, “CloudCompare Open Source Project.”
- [12] EDF, “CloudCompare Command Line Mode.”
- [13] Russell, J. and Cohn, R., *OpenShot Video Editor*. Book on Demand, 2012.
- [14] Mur-Artal, R. and Tardós, J. D., “ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [15] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. R., “Orb: An efficient alternative to sift or surf.”
- [16] Kalman, R., *A New Approach to Linear Filtering and Prediction Problems*, vol. 82. jan 1960.
- [17] Schröder, P., “Digital geometry processing,” *Course Notes# 50 for Siggraph*, vol. 6, pp. 6041–4, 2001.

A SOFTWARE EMPLEADO

A.1. Blender

Para la generación del espacio 3D se ha decidido usar el software Blender [10]. Aunque este es el software que se ha usado en la realización de este proyecto, no fue la idea original. En una primera aproximación se optó por usar Unity pero las limitaciones a la hora de proyectar la trayectoria de la cámara acabó descartándolo.

Uno de los aspectos positivos de trabajar con Blender, es su licencia de software libre que lo convierte en la herramienta perfecta para trabajar en el ámbito académico. Esta herramienta destaca por la sencillez de su interfaz, cosa que facilita en gran medida el trabajar con ella y además el gran soporte que tiene para realizar scripting permite una interacción código-escena que hace viable el proyecto. Además presenta una funcionalidad muy útil, que permite importar directamente un fichero en formato Python que configura el entorno con el que se va a trabajar aparte de la gran cantidad de configuraciones que presenta a la hora de automatizar el renderizado de la animación que se acaba de crear.

A.2. CloudCompare

CloudCompare [11] es una herramienta con GUI que permite interactuar con las nubes de puntos de manera muy sencilla. Otro punto a favor de este es que es completamente software libre por lo que se alinea correctamente con el objetivo del proyecto que es que sea realizado íntegramente con software sin restricciones de copyright. Esta herramienta se ha usado por que permite realizar un ICP de forma extremadamente sencilla y además la implementación particular en este software es altamente eficaz que otras opciones probadas, y es necesaria esta eficiencia ya que en las pasadas finales del algoritmo donde ya se han acumulado muchas nubes de puntos en una nube conjunta, resulta imprescindible que el software sea capaz de manejarlo con eficacia. Por último, la posibilidad de trabajar con esta herramienta mediante la línea de comandos [12] lo convierte en la herramienta perfecta para visualizar los puntos de forma sencilla. Además provee de todas las funcionalidades que se necesitan para la automatización del proceso. Esta línea de comandos permite acceder a todas las funcionalidades base de la misma manera que se hacen mediante la interfaz gráfica a excepción del acceso a los plugins.

A.3. Openshot

Openshot [13], es un programa de edición de vídeo que permite realizar la unión de los dos fragmentos de vídeo. Es un software completamente gratuito y con disponibilidad para las plataformas Windows, Mac OS y Ubuntu. Se ajusta perfectamente a las necesidades de este proyecto y aunque no contiene una gran cantidad de herramientas ni efectos que permitan hacer grandes ediciones presenta una interfaz con una gran cantidad de uso. El proceso para realizar la unión consta de la importación de ambos vídeos en pistas diferentes y aplicándole el efecto Chroma Key al vídeo realizado en Ubuntu. Esto permitiría la superposición de este último extrayéndole todo el fondo negro y dejando únicamente los elementos que desea el usuario.

A.4. ORB_SLAM2

Esta herramienta permite realizar el correcto análisis de la posición de la cámara en todo momento con un estrecho margen de error. Es una herramienta de código abierto disponible en GitHub [14], que permite realizar esta tarea tanto en vídeos Monoculares (vídeo grabado con una única lente), Estéreo (mediante dos lentes) o en RGB-D. Una de las virtudes de esta herramienta es su alta eficiencia ya que trabaja con threads independientes para las tareas que componen el sistema ya que son independientes. También desempeña unas políticas de puntos clave muy estrictas lo que se traduce directamente en un mejor rendimiento a nivel global de la parte del software destinada a realizar un correcto mapeo de las características fundamentales de las imágenes. Aunque no se ha explorado en este trabajo con otras posibilidades que ofrece esta tecnología, también presenta otras características como ser capaz de reconocer cual es la posición de la cámara en el circuito global si ya ha visto esa misma escena aunque presente modificaciones en la orientación de la cámara.

A.4.1. Pipeline

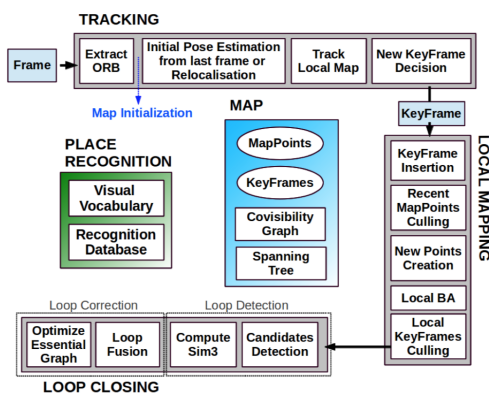


Fig. 16: Esquema del pipeline de ORB_SLAM2

ORB_SLAM2 provee de un sistema complejo para analizar la escena, este proceso no se realiza de forma monolítico sino que está compuesto de una serie de módulos con distintas funcionalidades. Los tres módulos se clasifican como Tracking, Local Mapping y Loop Closing. Estos tres módulos lejos de procesarse de forma secuencial se ejecutan en paralelo ya que son independientes entre ellos y por tanto pueden usarse threads independientes para cada uno de ellos. Este proceso va generando el mapa, es decir, el estado del sistema en tiempo real. La parte generada del mapa esta representada en forma de grafo (concretamente en forma de Spanning Tree para la construcción del grafo esencial). Simultáneamente se van construyendo las estructuras que permiten la representación de la escena mediante descriptores ORB y técnicas de clustering, que permiten realizar las tareas de reconocimiento de la localización.

Tracking Es en esta etapa en particular donde se analiza la imagen para encontrar el conjunto de características mediante descriptores de ORB [15] que la representa. Para realizar este proceso además se usan transformaciones de escala en la imagen y se producen segmentaciones en

regiones de tamaño fijo. Esto se realiza para poder encontrar características en todas las regiones posibles y evitar así que se produzcan grandes acumulaciones de estas características en regiones muy pequeñas. Después se realiza la estimación de la posición de la cámara dentro de un sistema de coordenadas mundo y se decide según una serie de criterios de obligado cumplimiento si este frame es o no un keyframe de la secuencia.

Local Mapping Aquí es donde se construye el conjunto de información que define la escena con los datos que están siendo extraídos por el proceso de tracking. También es donde se va construyendo los grafos de posición de la cámara y se van generando los conjuntos de representación de estas escenas. Aparte de esto, también se produce el proceso de filtrado de los nuevos keyframes generados y se generan las representaciones de los puntos en el espacio 3D de los puntos visualizados en este subconjunto de frames actual. Se realizan al mismo tiempo otras técnicas de optimización que tienen el objetivo de reducir el impacto de los errores acumulados a lo largo de todo el proceso.

Loop Closing En este último módulo se analiza si la trayectoria descrita por el vehículo se ha cerrado. Esto quiere decir que el vehículo ha vuelto a un lugar por el que ha pasado previamente. La estrategia de cerrar bucles es muy común ya que permite regular el nivel de error que se va produciendo a lo largo del tiempo y así reducirlo mejorando el aspecto general de la escena.

A.4.2. ORB

ORB (Oriented FAST and Rotated BRIEF) [15] parte de la idea planteada por los descriptores BRIEF presentando invariancia a rotación y alta resistencia al ruido. Nace con la idea de substituir a otras técnicas que habían sido la referencia durante mucho tiempo cómo SIFT o SURF, pero que presentan problemas en temas de eficiencia y bajos niveles de detección de características respectivamente. Otro punto a favor del uso de ORB, es que a diferencia de las alternativas mencionadas es de uso libre haciéndolo ideal para trabajar en el ámbito académico.

A.4.3. SLAM

SLAM o Simultaneous Localisation and Mapping, es una técnica dónde simultáneamente se realiza un mapeado del entorno y se realiza una reconstrucción de la trayectoria que ha seguido la cámara. La generación del entorno se realiza de forma incremental y es considerado el problema más importante en el campo de la robótica móvil. Los avances en este campo han sido notables en la última década y incluso puede llegarse a considerarse como un problema solventado a día de hoy. Las técnicas usadas tienen su fundamento en las técnicas probabilísticas, el motivo de esto es que la información percibida a través de los sensores del vehículo no es perfecta ya que de ser así se podría conseguir reconstruir el estado del sistema en todo momento con absoluta precisión. El principio que funciona como fundamento para este tipo de aproximaciones probabilísticas es la regla de Bayes que permite establecer una probabilidad en la sucesión de

un evento si ha ocurrido otro de forma previa (Filtros de Kalman [16]).

B ALGORITMOS Y OTROS CONCEPTOS

B.1. LIDAR

Para la captura del escenario en las secuencias de conducción se requiere de un hardware especial conocido como LIDAR o Laser Imaging Detection and Ranging basado en el uso de láser para poder crear un modelo tridimensional con el aspecto que presenta el objeto escaneado. Para calcular las distancias asociadas a cada uno de los puntos donde rebota el láser pulsado se utiliza el tiempo de vuelo del láser desde que es emitido hasta que se vuelve a captar en el sensor. En el caso particular de los vehículos autónomos se suele usar un tipo de LIDAR conocido como omnidireccional que proyecta en todas las direcciones los pulsos del láser. En el caso particular de la configuración que va instalada en el vehículo donde se han capturado los datos del dataset KITTI, se trata del modelo HDL-64E de la compañía Velodyne.

B.2. Point Clouds

Una nube de puntos o Point Clouds en inglés [17] es una representación tridimensional de un conjunto de vértices que forman la geometría de un objeto que se ha obtenido mediante técnicas de luz pulsada como el LIDAR. Estos datos obtenidos pueden ser guardados en muchos formatos distintos aunque aquellos que vienen dados por el dataset aquí usado es un vector de cuatro componentes donde sus tres primeras indican la posición que ocupa en el espacio y la cuarta indica el nivel de reflectancia asociado a ese punto. La nube de puntos capturada en un único frame resulta ineficiente a la hora de realizar una reconstrucción completa de la escena, pero el interpolar un gran número de ellos reporta un resultado final mucho más preciso. Existen numerosos formatos de Point Clouds muchos de ellos pertenecientes a distintos fabricantes y la mayoría convertibles entre ellos. Además este formato presenta gran flexibilidad ya que es posible guardar dentro del mismo formato no únicamente la posición de los puntos sino además las normales asociadas o el color que tiene cada punto. El aspecto que presentan los ficheros equivalen a una línea para cada uno de los puntos de la nube con tantos campos como información adicional se requiera aparte de las coordenadas x , y , z . Es preferible almacenar las nubes de puntos en su variante binaria ya que ocupan mucho menos espacio que su correspondencia en formato ASCII.

B.3. ICP

Al elegir usar las nubes de puntos que obtenemos mediante el uso de LIDAR, necesitamos disponer de una forma de reconstrucción completa de la escena. El formato que disponen muchos datasets está formado por los datos LIDAR asociados a cada uno de los frames del vídeo original. El problema es convertir cada uno de estos conjuntos de puntos independientes, en una nube de puntos que los agrupe todos.

Con el objetivo de realizar esto, necesitamos usar herramientas que permitan establecer una correspondencia entre los distintos conjuntos de puntos y que permita realizar transformaciones como la translación y la rotación para que el alineado sea correcto. Con este propósito se hace uso de un algoritmo conocido como ICP o Iterative Closest Point [5]. Este algoritmo se basa en establecer uno de los dos conjuntos de puntos como referencia y mantenerlo fijo, mientras se realizan transformaciones sobre la otra nube y se evalúa la eficacia de este alineamiento. Esta evaluación se suele basar en una estimación de las distancias de los puntos por parejas. El punto que diferencia esta estrategia de otras, es que en aproximaciones previas a este tipo de problemas se requería de la introducción de las parejas de forma manual. Se debían seleccionar 4 puntos de forma arbitraria (2 para cada nube) para poder establecer la correspondencia. En cambio, con el uso de ICP las correspondencias entre ambas nubes se realizan de forma automática en cada una de las iteraciones del algoritmo haciéndolo más sencillo de usar y el método ideal para trabajar en problemas que requieran de la automatización.

Los pasos que realiza el algoritmo pueden verse resumidos en los siguientes puntos:

1. Se busca la correspondencia de cada punto con el punto más próximo en la nube que se usa como base.
2. Estimación de la rotación y traslación mediante la métrica MSE. Aquí además se pueden implementar algunas estrategias para mejorar el rendimiento global del algoritmo como realizar estrategias de ponderación o rechazar outliers que se hayan podido detectar.
3. Se aplican las transformaciones obtenidas a la nube de puntos que queríamos alinear.
4. Se asocian de nuevo los puntos de ambas nubes y se regresa al primer paso.

C PLANIFICACIÓN

C.1. Metodología

La metodología de desarrollo que se ha seguido está basada en la realización de distintos bloques de forma secuencial. Cada uno de estos bloques está pensado para desarrollar una tarea particular y bien diferenciada del resto. Las primeras iteraciones de este proyecto están enfocadas a documentarse sobre el tema que se va a desarrollar a lo largo del trabajo. Debido al calendario ajustado es muy importante realizar un proceso de documentación exhaustivo al principio para poder evaluar las mejores estrategias. La construcción de los módulos se ha realizado de forma secuencial debido a la naturaleza del trabajo. En este proyecto en particular es imprescindible todos los módulos estén desarrollados para que se pueda llegar a observar un resultado. Por este mismo hecho que el orden del desarrollo que se ha seguido ha sido arbitrario a excepción del proceso de documentación previo. Cada uno de estos módulos son tareas independientes que deben estar completadas, por tanto

ha sido imprescindible cumplir con las fechas con la mayor precisión posible para poder completar el proyecto sin dejar nada por terminar y que no se pudiera realizar con éxito. También se ha asignado una gran cantidad de tiempo a evaluar las herramientas y necesidades que van a ser usadas en el desarrollo de los módulos ya que de estas depende directamente la calidad final del proyecto. A continuación se muestra un diagrama de Gantt que indica cual va a ser el orden de desarrollo de cada uno de los módulos y además se muestra la duración estimada de cada uno de ellos. Aunque no se han mostrado en este diagrama de Gantt, también se han desarrollado de forma paralela las distintas entregas que se han producido a los módulos que se tuvieran que desarrollar esa semana. Aunque se ha trabajado más esas semanas los módulos se han completado de la forma que se habían establecido al principio.



Fig. 17: Diagrama de Gantt del desarrollo del proyecto.

Aunque este es el estado final del proyecto, han surgido algunas modificaciones respecto a las versiones previas del informe. Las tareas se han mantenido ya que siguen siendo necesarios cada uno de los módulos, pero si que se han introducido cambios en cada uno de ellos con el objetivo de adaptarlos a las nuevas condiciones de desarrollo que se han ido produciendo. Dicho de otra manera, se han cambiado las estrategias para afrontar las distintas tareas que debían de realizarse ya que se han evaluado distintas alternativas y algunas han acabado por substituir las ideas iniciales. Ha sido necesario estar en contacto de forma permanente con el tutor para mantenerlo informado acerca de las estrategias que se iban a usar en los distintos módulos para valorarlas con él ya que tiene una mayor experiencia en el campo. En el último periodo del desarrollo de este proyecto se ha realizado de manera simultánea la elaboración de este documento final al mismo tiempo que se han explorado mejoras que se pudieran aplicar al proyecto. También se han explorado líneas futuras de trabajo sobre las que trabajar en el caso de seguir desarrollando este proyecto para mejorar la calidad una vez se ha completado con éxito. También en este último intervalo destinado a las mejoras se ha reescrito el código para hacerlo más legible y se ha añadido un fichero de configuración para facilitar su uso.

D EJEMPLOS DE SECUENCIAS



Fig. 18: Muestra del resultado obtenido en una secuencia de vídeo donde se ha insertado un vehículo en movimiento.



Fig. 19: Muestra del resultado obtenido en una secuencia de vídeo donde se ha insertado una señal de tráfico en el lado derecho de la carretera.